Solución:

Ejercicio 1:

```
% Escribir aquí el código
% Volcar el valor aproximado de cos(x) obtenido
% Despues de dar el comando >>format long, comparar el resultado obtenido con el resultado comando matlab cos()
% Cuántas cifras significativas de precisión habéis obtenido
```

```
format long
>> N=5;n=0:N;x=0.5;y=((-1).^n).*(x.^(2*n));z=factorial(2*n);v=sum(y./z),cos(0.5)
v =
  0.87758256188986
ans =
  0.87758256189037
```

Se han obtenido 10 cifras significativas.

```
% Insertad aquí el código de la función
```

```
function [cos_aprox,E_rel]=coseno(x,n)
k=0:n;y=((-1).^k).*(x.^(2*k));z=factorial(2*k);cos_aprox=sum(y./z);E_rel=abs(cos_aprox-cos(x))/abs(cos(x));
return
```

```
% Volcar los resultados de la ejecución de la función
```

```
>> [cos_aprox,E_rel]=coseno(0.5,5)
cos_aprox =
  0.87758256188986
E_rel =
   5.800448628948748e-013
```

Sea x= $\pi/4$, ejecutar la función coseno para n=5, 6, 7, …

```
% Volcar el resultado de las ejecuciones
>> [cos_aprox,E_rel]=coseno(pi/4,7)
cos_aprox =
  0.70710678118655
E_rel =
   1.570092458683775e-015
>> [cos_aprox,E_rel]=coseno(pi/4,8)
cos_aprox =
  0.70710678118655
E_rel =
   1.570092458683775e-016
% Cual es el Error relativo mínimo
1.57 e-16
% ¿Por qué?
Es el error relativo en doble precisión.
% Cual es el primer n con el que se obtiene ese Error relativo mínimo.
n=8
% Volcar los resultados para ese n.
>> [cos_aprox,E_rel]=coseno(pi/4,8)
cos_aprox =
  0.70710678118655
E_rel =
   1.570092458683775e-016
```

Ejecutar la función coseno para x= 10π , ¿es adecuada numéricamente la estimación del coseno para x= 10π ?

```
% Contestar a la pregunta, justificando la respuesta.
```

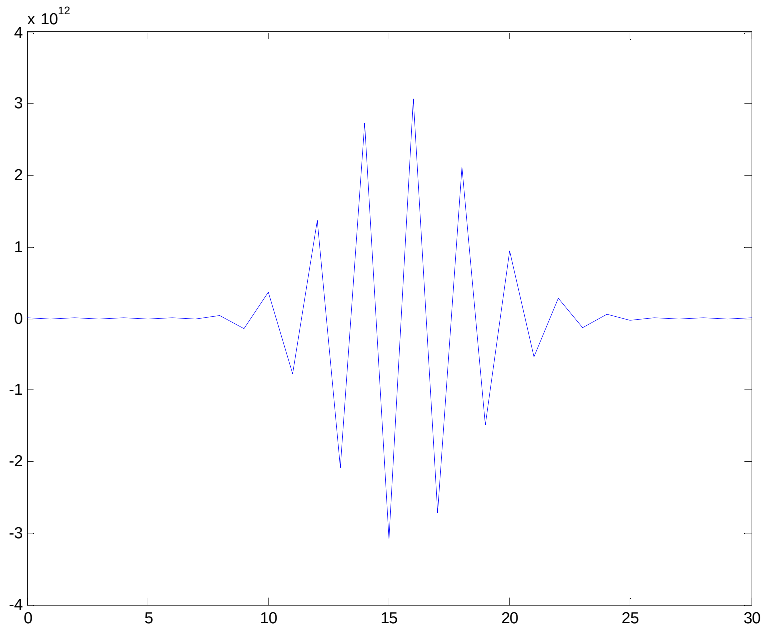
```
Para x=10pi.
>> [cos_aprox,E_rel]=coseno(10*pi,7)
cos_aprox =
  -8.770067202072704e+009
E_rel =
   8.770067203072704e+009
>> [cos_aprox,E_rel]=coseno(10*pi,17)
```

```
cos_aprox =
  -1.224057949028887e+012
E_rel =
   1.224057949029887e+012
>> [cos_aprox,E_rel]=coseno(10*pi,27)
cos_aprox =
  -7.472797024476643e+008
E_rel =
   7.472797034476643e+008
```

Todos los E_rel son mayores de 1e+8, luego NO es una buena estimación numérica para x=10pi.

El inconveniente numérico radica en que, para valores de x grandes, los términos de la serie son de una magnitud muy grande (del orden de 10^12).

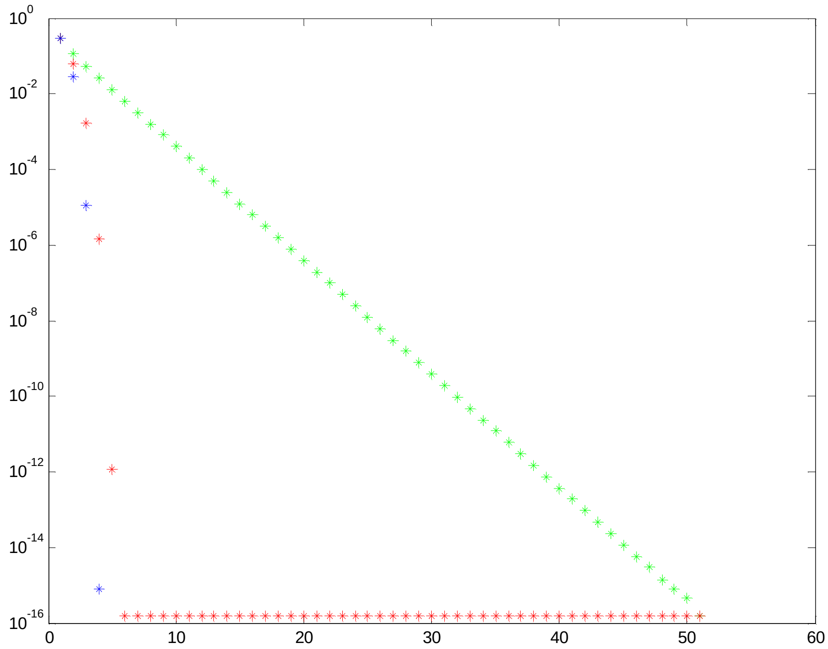
```
N=30;n=0:N;x=10*pi;y=((-1).^n).*(x.^(2*n));z=factorial(2*n);w=y./z;
 plot(n,w)
```



Ejercicio 2:

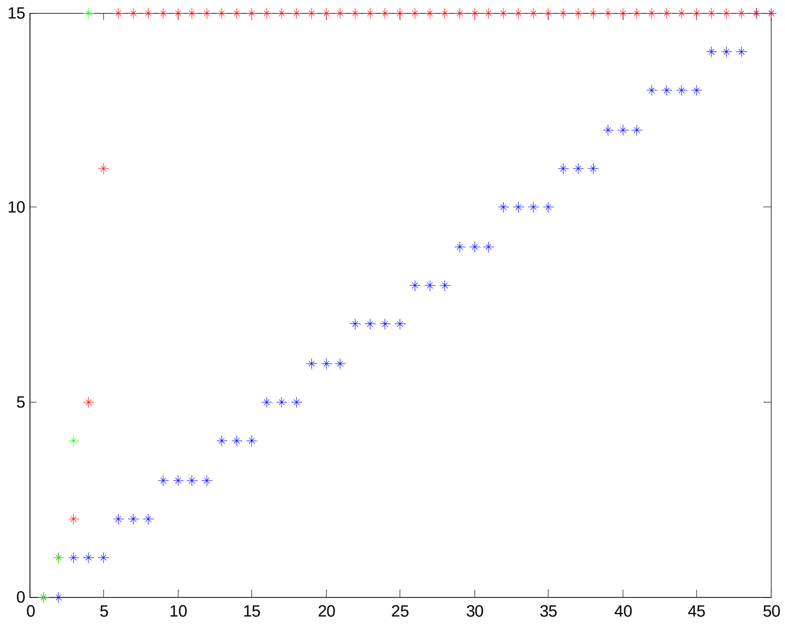
```
% Insertar aquí el código pedido
% Volcar aquí el contenido de las 3 tablas de errores relativos
% Insertar aquí la gráfica
```

```
N=51;
x=NaN(1,N);y=NaN(1,N);z=NaN(1,N);
x(1)=1;y(1)=1;z(1)=1;
for n=1:N-1
   x(n+1)=3*x(n)/4+1/(2*x(n));
   y(n+1)=y(n)/2+1/y(n);
   z(n+1)=3*z(n)/8+3/(2*z(n))-1/(2*z(n)^3);
end
   E_rel_x=abs(sqrt(2)-x)/sqrt(2);
   E_rel_y=abs(sqrt(2)-y)/sqrt(2);
   E_rel_z=abs(sqrt(2)-z)/sqrt(2);
   k=N;
     semilogy([1:k],E_rel_x(1:k),'g*',[1:k],E_rel_y(1:k),'r*',[1:k],E_rel_z(1:k),'b*')
```



- Evaluar el número de cifras significativas obtenidas con cada una de las 3 sucesiones. Representarlas (utilizando los mismos colores anteriores), respecto de n.

```
% Insertar aquí el código y la gráfica
n_cifras_x=floor(-log10(E_rel_x));n_cifras_y=floor(-log10(E_rel_y));n_cifras_z=floor(-log10(E_rel_z));
plot([1:k],n_cifras_x,'b*',[1:k],n_cifras_y,'r*',[1:k],n_cifras_z,'g*')
```



- ¿Cuál es el máximo de número de cifras significativas de precisión que se han obtenido?.
- ¿Cuántas iteraciones son necesarias para que la sucesión x_n alcance el máximo número de cifras significativas?, ¿la sucesión y_n?, ¿y la

sucesión z_n?. ¿Cuál de las 3 sucesiones es la más rápida? ¿cuál es la más lenta?.

```
% El máximo número de cifras significaticas es 15.
La sucesión x_n necesita 50 iteraciones para alcanzar las 15 cifras
significativas, la sucesión y_n necesita 5 iteraciones y la sucesión z_n
únicamente 3. La sucesión más rápida es la z_n y la mas lenta la x_n.
```